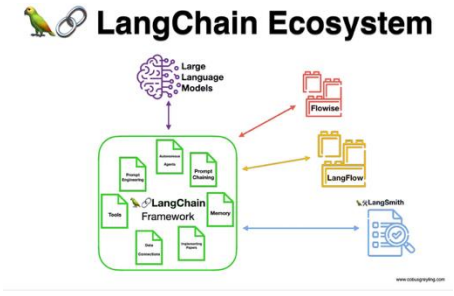


LangChain: Orquestrando LLMs para Aplicações Inteligentes

Framework Essencial para Desenvolvimento com Grandes Modelos de Linguagem

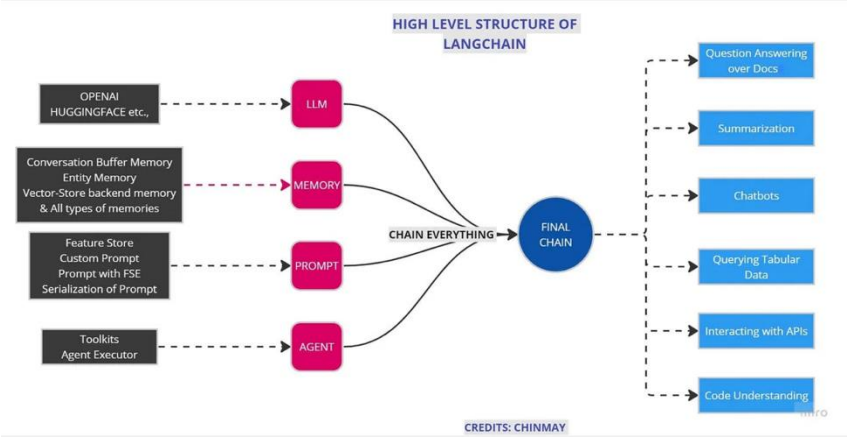
LLMs: Potencial Imenso, Desafios Complexos na Aplicação

Grandes Modelos de Linguagem (LLMs) oferecem capacidades sem precedentes, mas apresentam desafios significativos para implementação prática:

Desafios Críticos

- Complexidade de integração com sistemas existentes
- Gerenciamento de contexto e memória limitados
- Necessidade de orquestração entre múltiplos componentes
- Dificuldade em manter consistência nas respostas

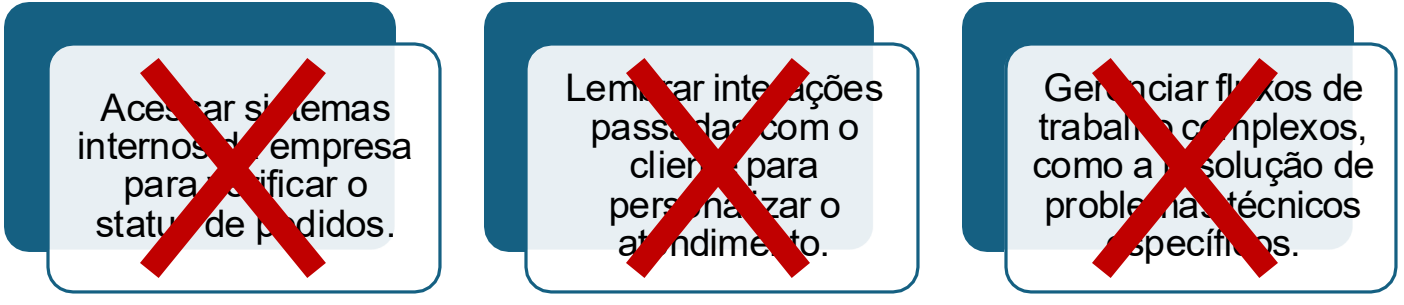
A implementação eficiente de LLMs requer uma abordagem estruturada para superar estas limitações e maximizar seu potencial em aplicações reais.



Assistente de Atendimento ao Cliente Utilizando Linguagem Natural

- ✓ Fornecer informações sobre produtos
- ✓ Prazos de Entrega
- ✓ Fornecer Suporte Técnico
- ✓ Status de Pedidos

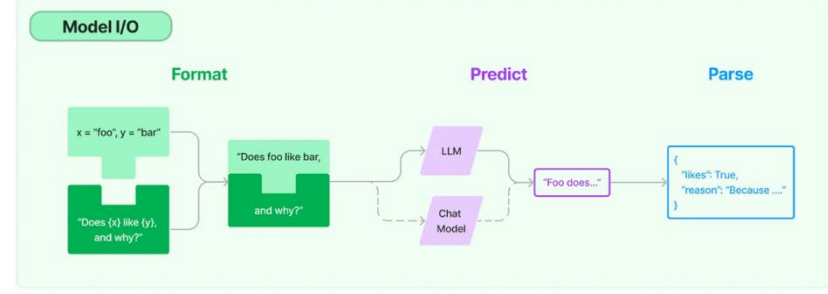
LLM


Acessar sistemas internos da empresa para verificar o status de pedidos.
Lembrar interações passadas com o cliente para personalizar o atendimento.
Gerenciar fluxos de trabalho complexos, como a resolução de problemas técnicos específicos.

LangChain Simplifica Desenvolvimento de Aplicações com LLMs

LangChain é um framework de orquestração que facilita a integração de LLMs em aplicações:

"LangChain é um framework de código aberto para desenvolvimento de aplicações baseadas em grandes modelos de linguagem (LLMs), que simplifica a integração e orquestração desses modelos."



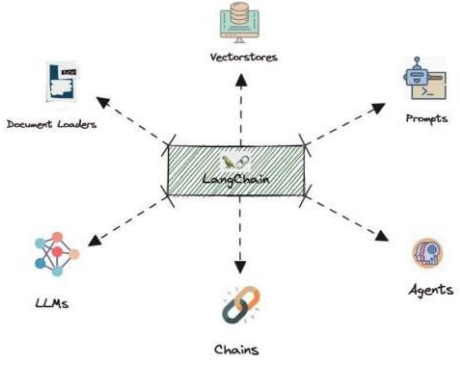
Benefícios Fundamentais

- Abstração de complexidade dos LLMs
- Modularidade e componentes reutilizáveis
- Integração simplificada com fontes externas
- Orquestração eficiente de fluxos de trabalho

LangChain

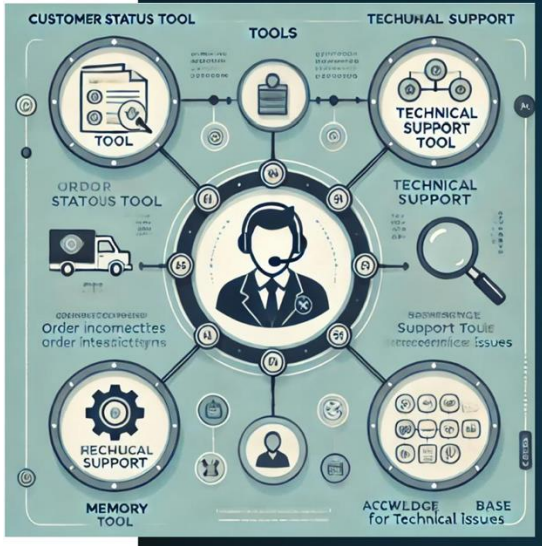
- É uma biblioteca de código aberto projetada para facilitar a integração de LLMs com várias fontes de dados e funcionalidades adicionais.
- Desenvolvido por Harrison Chase e lançado em outubro de 2022.
- Fornece uma sintaxe unificada que simplifica o uso de LLMs em diferentes contextos, como chatbots, análises de texto e sistemas de perguntas e respostas.

- LangChain torna mais fácil combinar LLMs com outras ferramentas e serviços.
- Além disso, permite que desenvolvedores criem soluções avançadas de processamento de linguagem natural de forma eficiente e escalável.
- É uma solução muito importante para deixar as LLMs “programáticas” e criar aplicações próprias.



Lanchain e LLM

- **Agente**
  - Gerencia a interação do usuário, coordenando a utilização das ferramentas para fornecer respostas precisas.
- **Ferramentas**
  - **Ferramenta de Status de Pedidos**: Conecta-se ao sistema de gerenciamento de pedidos da empresa.
  - **Ferramenta de Suporte Técnico**: Acessa uma base de conhecimento para resolver problemas técnicos.
- **Memória**
  - Armazena informações sobre interações passadas do cliente, como problemas anteriores e preferências de produtos.
- **Chains**
  - Cria uma sequência de passos para resolver uma consulta complexa.



Planejador de Viagens Personalizado

- Permitir pesquisa de viagens
- Sugerir roteiros
- Comparar preços
- Salvar preferências do usuário
- Realizar reservas, cancelamentos e pagamentos

Acessar informações em tempo real sobre disponibilidade e preços de voos e hotéis.

Integrar dados de múltiplas fontes, como clima, eventos locais e atividades.

Manter um contexto contínuo para personalizar recomendações baseadas em interações passadas.

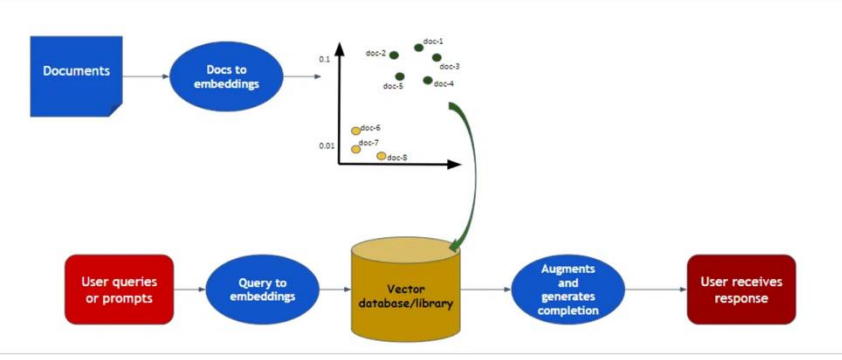
Lanchain e LLM

1. **Agente**
    1. Gerencia a interação do usuário, coordenando as diversas ferramentas para planejar a viagem.
2. **Ferramentas**
    1. **Ferramenta de Pesquisa de Voos e Hotéis**: Conecta-se a APIs de terceiros, como Skyscanner e Booking.com.
    2. **Ferramenta de Clima**: Conecta-se a uma API de clima, como OpenWeatherMap.
    3. **Ferramenta de Eventos Locais**: Conecta-se a APIs de eventos, como Eventbrite.
3. **Memória**
    1. Armazena informações sobre as preferências e interações passadas do usuário para personalizar futuras interações.
4. **Chains**
    1. Cria uma sequência de passos para planejar a viagem de maneira eficiente.
5. **Conexões com APIs**
    1. Facilita a comunicação com diversas APIs externas para buscar dados em tempo real.

LangChain: Futuro Promissor na Orquestração de IA

O LangChain estabeleceu-se como ferramenta essencial para desenvolvimento com LLMs:

- Simplifica a complexidade de integração de LLMs
- Oferece componentes modulares e reutilizáveis
- Evolui constantemente com novas ferramentas
- Possibilita aplicações de IA mais sofisticadas



Perspectivas Futuras

O ecossistema LangChain continuará expandindo, com foco em confiabilidade, escalabilidade e facilidade de uso para aplicações de produção.

LangChain: Habilitando Inovação em Diversos Cenários

O LangChain possibilita uma ampla gama de aplicações:

Sistemas de Q&A

Respostas baseadas em documentos específicos

Chatbots Avançados

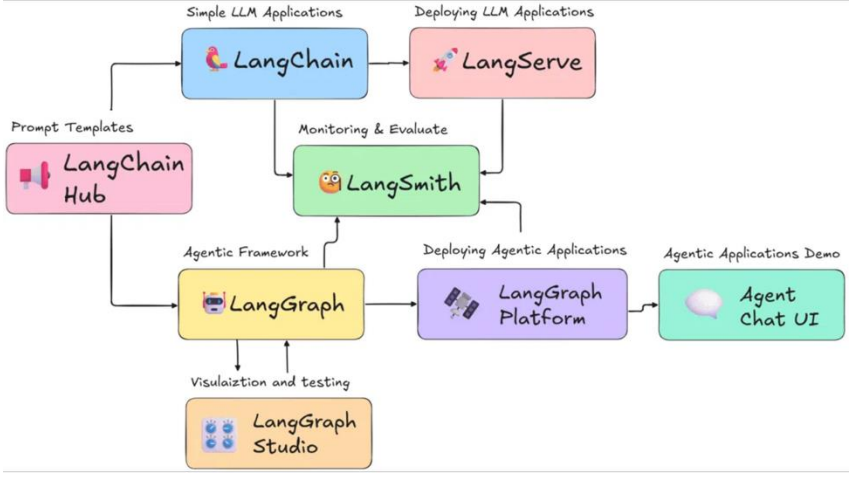
Assistentes com memória e personalidade

Agentes Autônomos

Sistemas que tomam decisões e executam ações

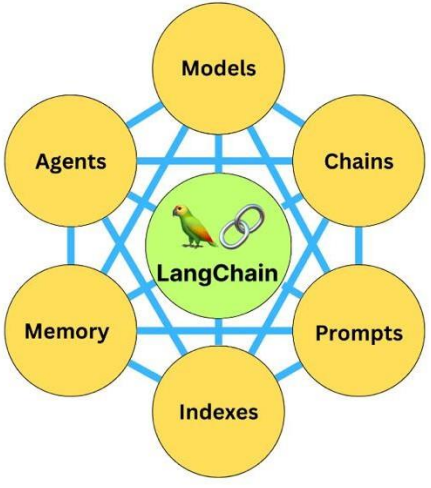
Análise de Documentos

Extração e sumarização de informações



LANGCHAIN – PRINCIPAIS COMPONENTES

- **Modelos:** Oferece uma interface padrão para interações com uma ampla gama de LLMs.
- **Prompts:** Ferramentas para simplificar a criação e tratamento de prompts dinâmicos.
- **Chains** (Corrente, Cadeia ou Sequencia): Interface padrão para encadear LLMs em aplicações complexas, permitindo a ligação entre múltiplos modelos ou outros módulos especializados.
- **Memória:** Módulos que permitem o gerenciamento e alteração de conversas anteriores, essencial para chatbots que precisam relembrar interações passadas para manter coerência.
- **Agentes:** Equipados com um kit de ferramentas abrangente, podem escolher quais ferramentas usar com base nas informações do usuário.
- **Índices:** Métodos para organizar documentos (que contém dados proprietários, por exemplo) de forma a facilitar a interação eficaz com LLMs.

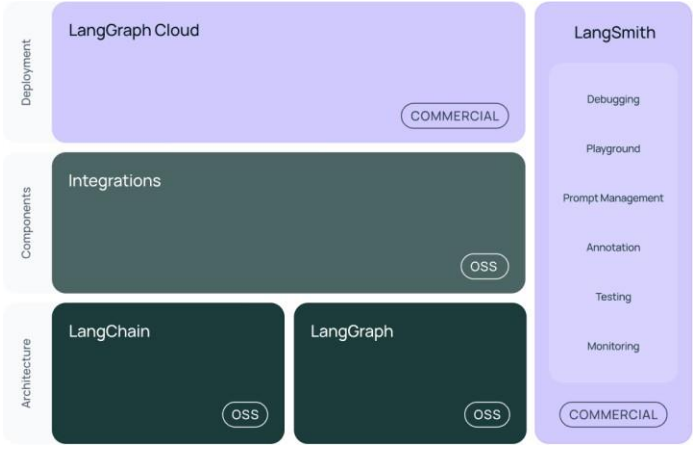


Créditos da imagem: ByteByteGo

ECOSSISTEMA LANGCHAIN

- langchain-core: Abstrações básicas e LangChain Expression Language (LCEL).
- langchain-community: Integrações de terceiros. Pacotes parceiros (por exemplo, langchain-openai, langchain-anthropic, etc.): Algumas integrações foram divididas em seus próprios pacotes leves que dependem apenas do langchain-core.
- langchain: Chains, Agentes e Estratégias de Retrieval que compõem a arquitetura cognitiva de uma aplicação.
- LangGraph: Para construir aplicações robustas e com estado para múltiplos atores com LLMs, modelando etapas como arestas e nós em um gráfico. Integra-se perfeitamente com LangChain, mas pode ser usado sem ele.
- LangServe: Para implementar chains do LangChain como APIs REST.
- LangSmith: Uma plataforma para desenvolvedores que permite depurar, testar, avaliar e monitorar aplicações LLM.

OOS = Open Source Software



https://python.langchain.com/v0.2/docs/introduction/

LANGCHAIN – MODELOS

- Uma das principais vantagens do LangChain é que ele permite trabalhar facilmente com diversos modelos. Alguns modelos são melhores para determinadas tarefas ou oferecem um melhor custo-benefício. Portanto, você provavelmente irá explorar diferentes modelos durante seus testes.
- O LangChain tem integrações com muitos provedores de modelos (OpenAI, Cohere, Hugging Face, Anthropic, Google, etc.) e expõe uma interface padrão para interagir com todos esses modelos.
- No LangChain, ao trabalhar com modelos nós definimos o que é chamado de **LLM Wrapper**. Um wrapper é como uma "embalagem" que facilita a utilização dos Grandes Modelos de Linguagem em aplicações.
- Já a **LLM** em si funciona como o "cérebro" ou motor da aplicação, realizando o processamento de linguagem natural.

COMPONENTES - LLMS E CHAT MODELS

Componente - Chat Models

- Modelos de linguagem mais novos usam sequências de mensagens como entradas e retornam mensagens de chat como saídas.
- Esses modelos permitem atribuir funções distintas às mensagens, diferenciando entre IA, usuários e instruções do sistema.
- Embora trabalhem com mensagens de entrada e saída, os wrappers LangChain permitem que esses modelos recebam strings como entrada. Isso facilita o uso de modelos de chat no lugar de LLMs tradicionais.
- Quando uma string é passada como entrada, ela é convertida em uma HumanMessage e então processada pelo modelo.

Componente - LLMs

- Modelos de linguagem que recebem uma string como entrada e retornam uma string.
- Tradicionalmente, esses são modelos mais antigos (modelos mais novos geralmente são modelos de chat).
- Embora os modelos subjacentes trabalhem com string in / string out, os wrappers LangChain permitem que esses modelos recebam mensagens como entrada.
- As mensagens recebidas são formatadas em uma string antes de serem passadas para o modelo.LangChain não hospeda nenhum LLM, mas depende de integrações de terceiros.

COMPONENTES - LLMS E CHAT MODELS

- Ou seja: LLMs de texto puro de entrada/saída de texto tendem a ser mais antigos ou de nível mais baixo.
- Muitos modelos populares são mais bem usados como modelos de chat / bate-papo (chat completion models), mesmo para casos de uso que não sejam de chat.
- O LangChain prioriza o Chat Models por estar mais associado a seu uso com modelos mais modernos (pelo menos na versão atual).

MENSAGENS

Alguns modelos de linguagem pegam uma lista de mensagens como entrada e retornam uma mensagem. Existem alguns tipos diferentes de mensagens.

No LangChain, todas as mensagens têm uma propriedade `role`, `content` e `response_metadata`.

- A função (role) descreve quem está dizendo a mensagem (ex: human, system). LangChain tem diferentes classes de mensagem para diferentes funções.
- A propriedade conteúdo (content) descreve o conteúdo da mensagem, podendo ser: uma string (a maioria dos modelos lida com esse tipo de conteúdo); ou uma lista de dicionários (isso é usado para entrada multimodal, onde o dicionário contém informações sobre esse tipo de entrada e esse local de entrada)
- A propriedade *response_metadata* contém metadados adicionais sobre a resposta. Os dados aqui são frequentemente específicos para cada provedor de modelo. É aqui que informações como *log-probs* (probabilidades de log) e uso de token podem ser armazenadas.

Uso Básico

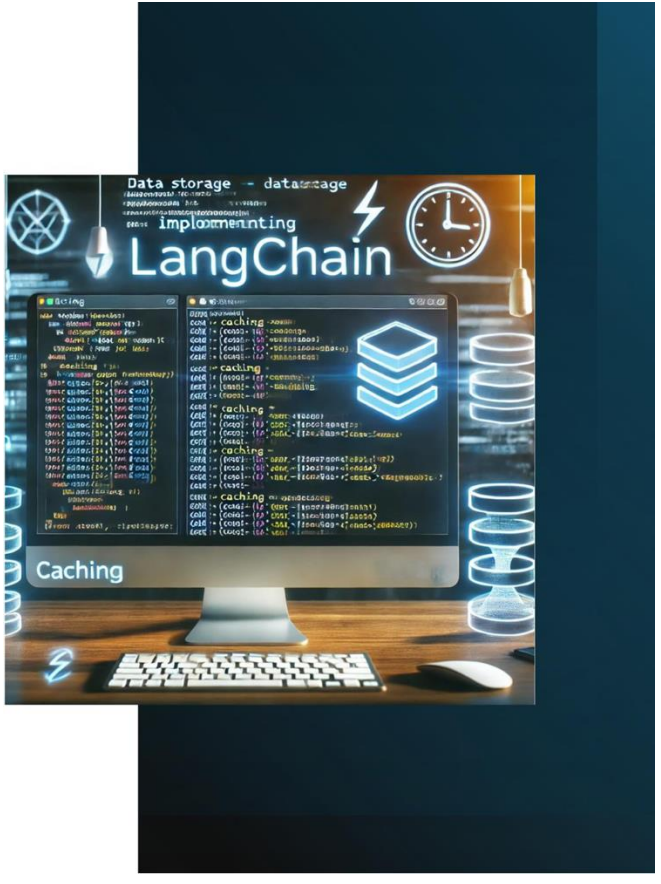
- Uso de um Modelo pelo Langchain
- Deve apresentar o mesmo resultado se usar a API do modelo diretamente

Caching

- Armazena resultados de consultas e cálculos para reutilização futura.
- Melhora a Performance: Reduz o tempo de processamento reutilizando resultados já calculados.
- Economia de Custos: Minimiza chamadas repetidas a APIs pagas.
- Eficiência: Diminui a carga no sistema, permitindo um uso mais eficiente dos recursos.

Caching no Langchain

- Memória
- Disco/SQLite
- Personalizado
- ...



Templates

- Modelos predefinidos para estruturar prompts e respostas de maneira consistente.
- Consistência: Garante que os prompts e respostas sigam um formato padrão.
- Eficiência: Facilita a criação de prompts complexos com menos esforço.
- Flexibilidade: Permite a personalização fácil para diferentes casos de uso, mantendo a estrutura básica intacta.

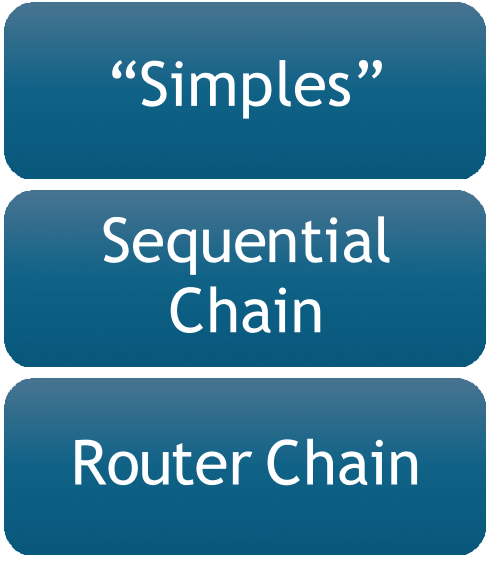
Exemplos

- Atendimento ao Cliente: Um template para responder perguntas frequentes, como "Qual é o status do meu pedido?" ou "Como posso retornar um produto?". Garante respostas rápidas e padronizadas para perguntas comuns.
- Marketing e Vendas: Templates para e-mails de vendas ou campanhas de marketing, como "Ofereça um desconto para novos clientes" ou "Anuncie um novo produto". Mantém a consistência da marca e facilita a criação de conteúdo atraente.

Chains

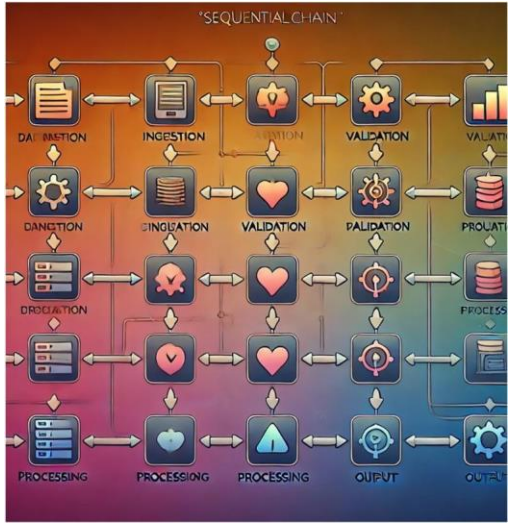
- Sequências de operações ou chamadas de API que são executadas de maneira encadeada para completar uma tarefa complexa.
- **Automatização de Processos**: Facilita a automação de fluxos de trabalho complexos.
- **Eficiência**: Reduz a necessidade de intervenção manual, permitindo que as tarefas sejam executadas de forma contínua e eficiente.
- **Flexibilidade:** Permite combinar diferentes operações e modelos para criar soluções personalizadas.

Exemplos



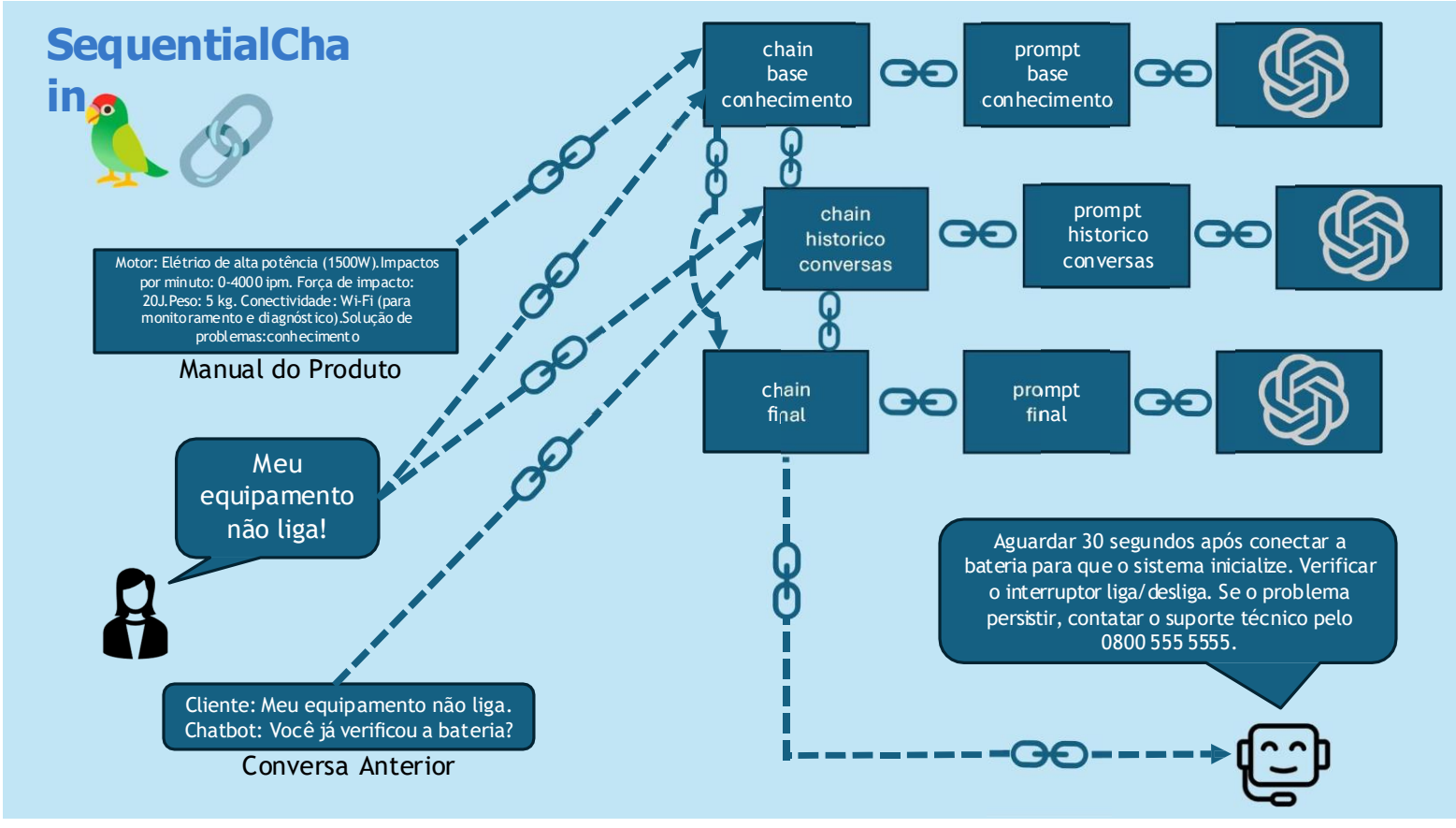
SequentialChain

- “n” etapas executadas em sequência
- A saída de uma etapa pode ser transformada
- Podemos definir quais serão as entradas de uma etapa



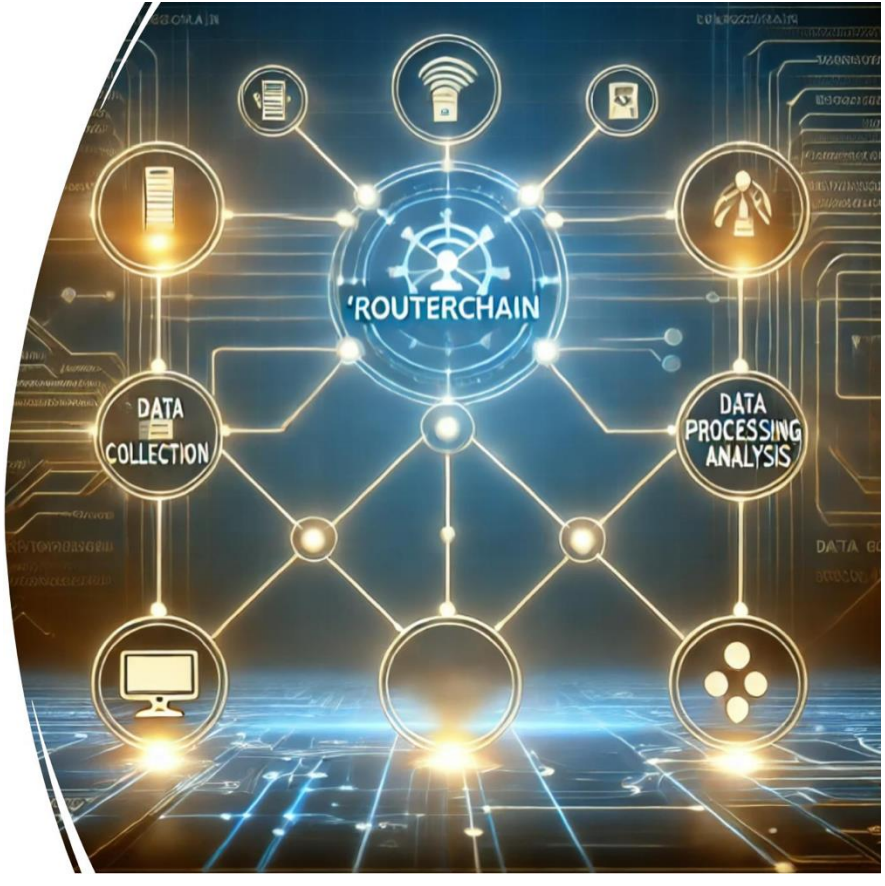
Atendimento ao Cliente

- Existe uma base de conhecimento
- Existe um histórico de conversas


SequentialCha
in
chain
base
conhecimento
prompt
base
conhecimento
chain
historico
conversas
prompt
historico
conversas
chain
final
prompt
final
Motor: Elétrico de alta potência (1500W). Impactos por minuto: 0-4000 ipm. Força de impacto: 20J. Peso: 5 kg. Conectividade: Wi-Fi (para monitoramento e diagnóstico). Solução de problemas: conhecimento
Manual do Produto
Meu equipamento não liga!
Cliente: Meu equipamento não liga.
Chatbot: Você já verificou a bateria?
Conversa Anterior
Aguardar 30 segundos após conectar a bateria para que o sistema inicialize. Verificar o interruptor liga/desliga. Se o problema persistir, contatar o suporte técnico pelo 0800 555 5555.

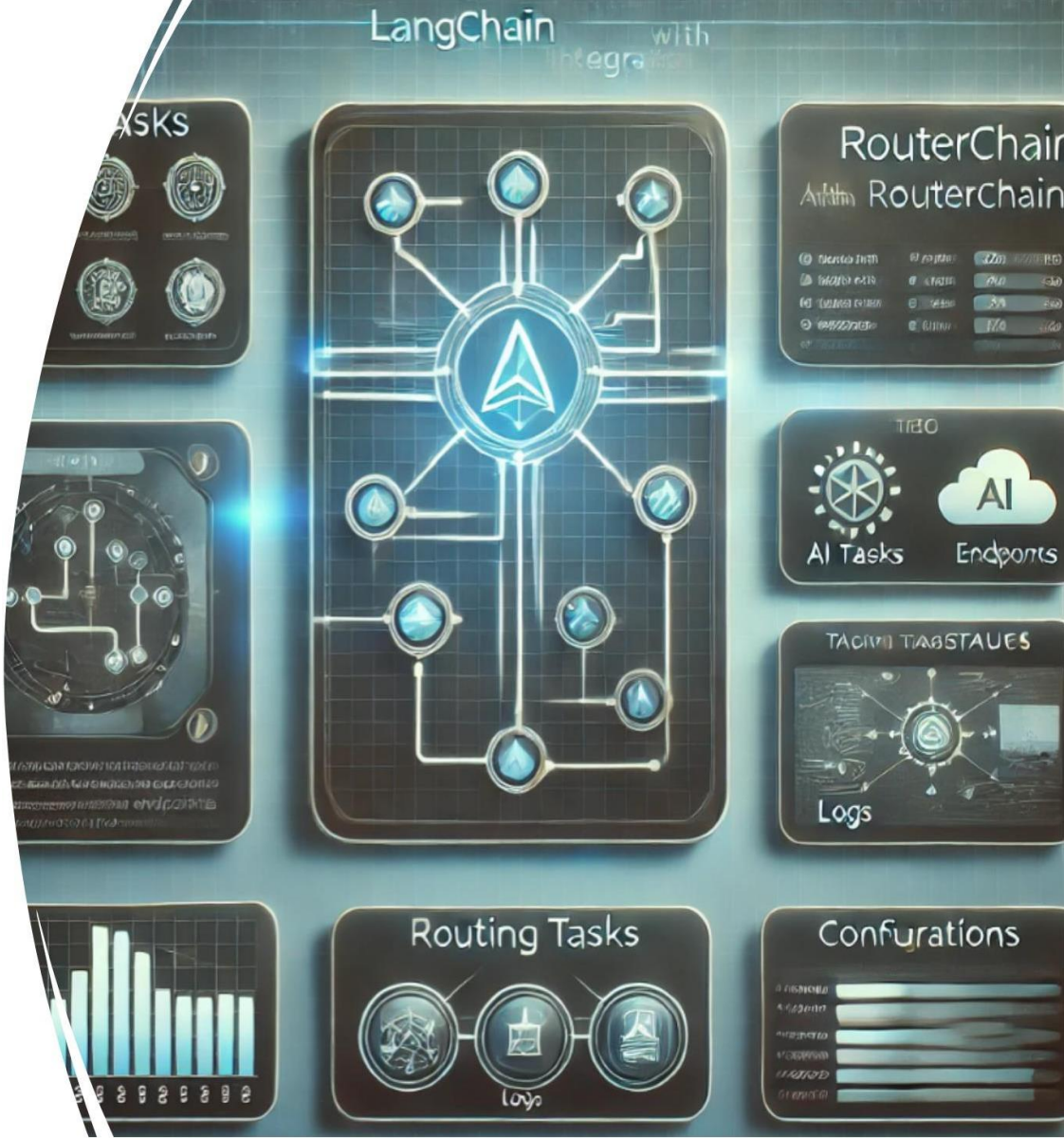
RouterChain

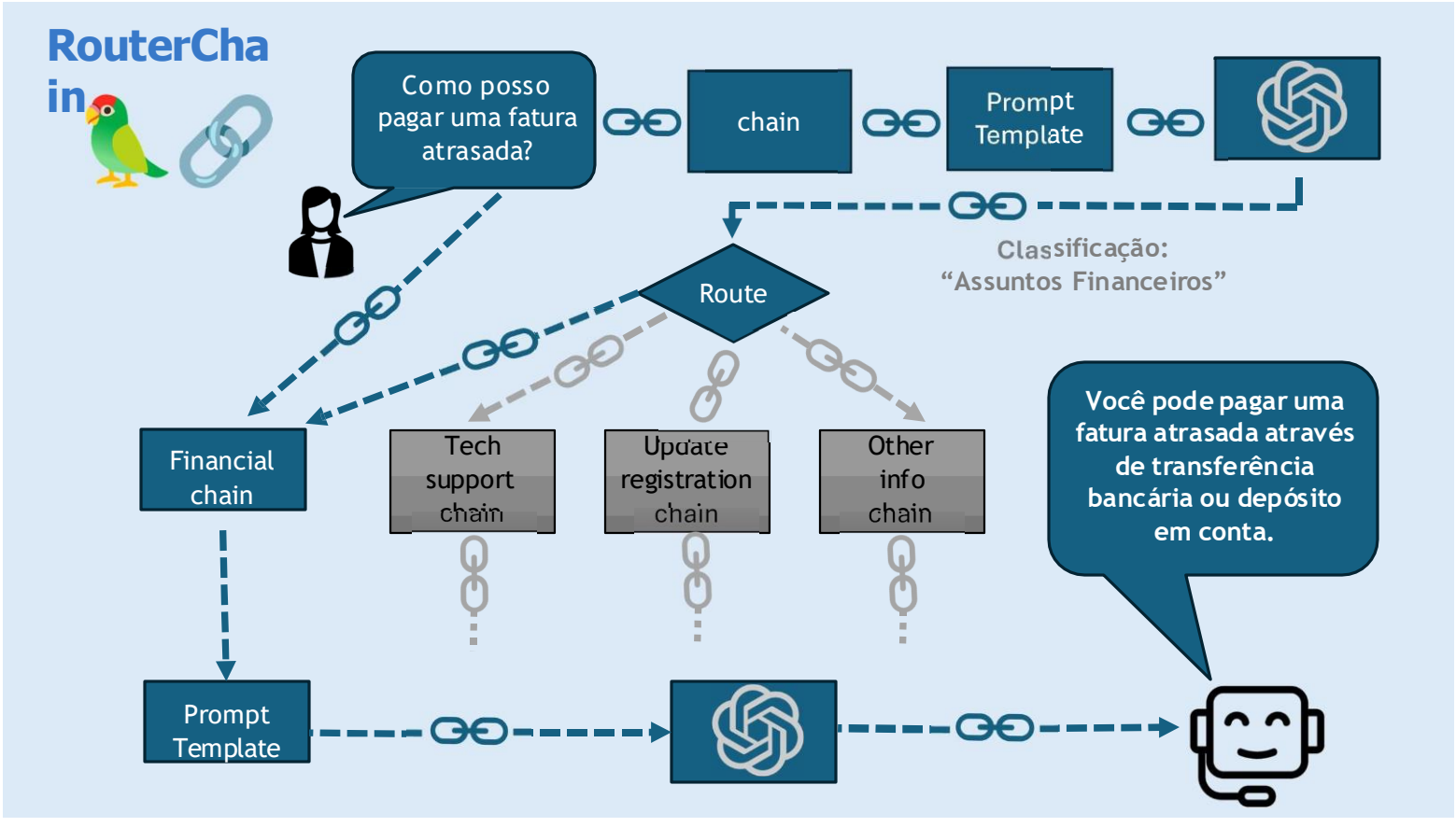
- Um “roteador” decide, com base na entrada do usuário, qual deve ser a etapa seguinte
- Cada etapa entre as opções pode ter diferentes templates, modelos etc. e seguirem fluxos diferentes.



Atendimento ao cliente:

- Dúvidas financeiras
- Suporte técnico
- Atualizar cadastro
- Outras opções




RouterChain
Como posso pagar uma fatura atrasada?
chain
Prompt Template
Classificação: "Assuntos Financeiros"
Route
Financial chain
Tech support chain
Update registration chain
Other info chain
Você pode pagar uma fatura atrasada através de transferência bancária ou depósito em conta.
Prompt Template

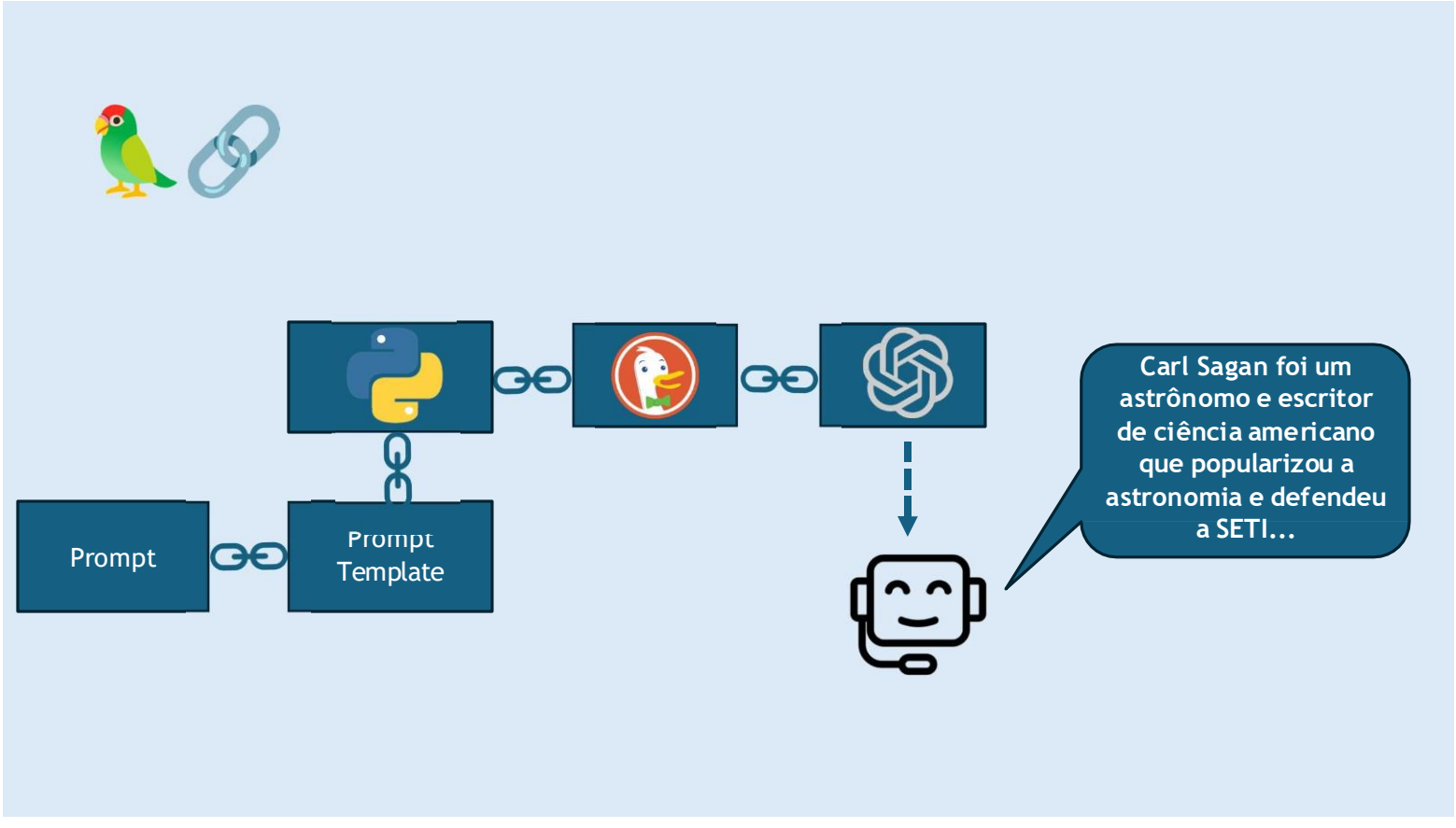
Tools

- Módulos ou componentes específicos que realizam tarefas determinadas, como busca de informações, análise de dados, ou interações com APIs externas.
- **Modularidade**: Permite a integração de diferentes funcionalidades de forma independente.
- **Reutilização**: Facilita a reutilização de componentes em diferentes projetos.
- **Expansibilidade**: Permite a adição de novas ferramentas para expandir as capacidades da aplicação conforme necessário.

Agents

- Componentes que realizam tarefas de forma autônoma, seguindo um conjunto predefinido de regras ou scripts.
- Operam de forma mais linear, executando tarefas com base em um fluxo de trabalho definido ou em uma sequência de comandos.

Agente de Pesquisa e Resumo



Agents

- Componentes que realizam tarefas de forma autônoma, seguindo um conjunto predefinido de regras ou scripts.
- Operam de forma mais linear, executando tarefas com base em um fluxo de trabalho definido ou em uma sequência de comandos.

React Agents

- Um subtipo de agents que respondem de maneira reativa a eventos ou mudanças no ambiente.
- São projetados para reagir dinamicamente a novas informações ou eventos em tempo real, ajustando seu comportamento com base nesses inputs.

Agente React Assitente Financeiro Pessoal